

Coarse abstractions make Zeno behaviours difficult to detect

F. Herbreteau and B. Srivathsan

Université de Bordeaux, IPB, Université Bordeaux 1, CNRS, LaBRI UMR5800
Bât A30, 351 crs Libération, 33405 Talence, France

Abstract. An infinite run of a timed automaton is Zeno if it spans only a finite amount of time. Such runs are considered unfeasible and hence it is important to detect them, or dually, find runs that are non-Zeno. Over the years important improvements have been obtained in checking reachability properties for timed automata. We show that some of these very efficient optimizations make testing for Zeno runs costly. In particular we show NP-completeness for the LU-extrapolation of Behrmann et al. We analyze the source of this complexity in detail and give general conditions on extrapolation operators that guarantee a (low) polynomial complexity of Zenoness checking. We propose a slight weakening of the LU-extrapolation that satisfies these conditions.

1 Introduction

Timed automata [1] are finite automata augmented with a finite number of clocks. The values of the clocks increase synchronously along with time in the states of the automaton and these values can be compared to a constant and reset to zero while crossing a transition. This model has been successfully used for verification of timed systems thanks to a number of tools [3, 6, 14].

Since timed automata model reactive systems that continuously interact with the environment, it is interesting to consider questions related to their infinite executions. An execution is said to be *Zeno* if an infinite number of events happen in a finite time interval. Such executions are clearly unfeasible. During verification, the aim is to detect if there exists a *non-Zeno* execution that violates a certain property. On the other hand while implementing timed automata, it is required to check the presence of pathological Zeno executions. This brings the motivation to analyze an automaton for the presence of such executions.

The analysis of timed automata faces the challenge of handling its uncountably many configurations. To tackle this problem, one considers a finite graph called the *abstract zone graph* (also known as simulation graph) of the automaton. This finite graph captures the semantics of the automaton. In this paper, we consider the problems of deciding if an automaton has a non-Zeno execution, dually a Zeno execution, given its abstract zone graph as input.

An abstract zone graph is obtained by over-approximating each zone of the so-called *zone graph* with an abstraction function. The zone graph in principle

could be infinite and an abstraction function is necessary for reducing it to a finite graph. The coarser the abstraction, the smaller the abstract zone graph, and hence the quicker the analysis of the automaton. This has motivated a lot of research towards finding coarser abstraction functions [2]. The classic maximum-bound abstraction uses as a parameter the maximal constant a clock gets compared to in a transition. A coarser abstraction called the LU-extrapolation was introduced in Behrmann et al. [2] for checking state reachability in timed automata. This is the coarsest among all the implemented approximations and is at present efficiently used in tools like UPPAAL.

It was shown in [12, 13] that even infinite executions of the automaton directly correspond to infinite paths in the abstract zone graph when one uses the maximum-bound approximation. In addition, it was proved that the existence of a non-Zeno infinite execution could be determined by adding an extra clock to the automaton to keep track of time and analyzing the abstract zone graph of this transformed automaton. A similar correspondence was established in the case of the LU-extrapolation by Li [11]. These results answer our question about deciding non-Zeno infinite executions of the automaton from its abstract zone graph. However, it was shown in [9, 10] that adding a clock has an exponential worst case complexity. A new polynomial construction was proposed for the case of the classic maximum-bound approximation. But, the case of the LU-extrapolation was not addressed.

In this paper, we prove that the non-Zenoness question turns out to be NP-complete for the LU-extrapolation, that is, given the abstract zone graph over the LU-extrapolation, deciding if the automaton has a non-Zeno execution is NP-complete. We study the source of this complexity in detail and give conditions on abstraction operators to ensure a polynomial complexity. To this regard, we extend the polynomial construction given in [9] to an arbitrary abstraction function and analyze when it stays polynomial. It then follows that a slight weakening of the LU-extrapolation makes the construction polynomial. In the second part of the paper, we repeat the same for the dual question: given an automaton's abstract zone graph, decide if it has Zeno executions. Yet again, we notice NP-completeness for the LU-extrapolation. We introduce an algorithm for checking Zenoness over an abstract zone graph with conditions on the abstraction operator to ensure a polynomial complexity. We provide a different weakening of LU-extrapolation that gives a polynomial solution to the Zenoness question.

Related work As mentioned above, the LU-extrapolation was proposed in [2] and shown how it could be efficiently used in UPPAAL for the purpose of reachability. The correctness of the classic maximum-bound abstraction was shown in [4]. Extensions of these results to infinite executions occur in [13, 11]. The trick involving adding an extra clock for non-Zenoness is discussed in [9]. For the case of checking existence of Zeno runs in timed automata, a bulk of the literature directs to [8, 5]. They provide a sufficient-only condition for the absence of Zeno runs. This is different from our proposed solution which gives a complete solution (necessary and sufficient conditions) by analyzing the abstract zone graph of the automaton.

Organization of the paper We start with the formal definitions of timed automata, abstract zone graphs, the Zenoness and Non-Zenoness problems in Section 2. Subsequently, we prove the NP-completeness of the non-Zenoness problem for the LU-extrapolation in Section 3. We then recall the construction proposed for non-Zenoness in [9] and extend it to a general abstraction operator giving conditions for polynomial complexity. Section 5 talks about the dual Zenoness problem and Section 6 concludes the paper with some perspectives.

2 Zeno-related Problems for Timed Automata

2.1 Timed Automata

Let $\mathbb{R}_{\geq 0}$ denote the set of non-negative real numbers. Let X be a set of variables, named *clocks* hereafter. A *valuation* is a function $\nu : X \mapsto \mathbb{R}_{\geq 0}$ that maps every clock in X to a non-negative real value. We denote the set of all valuations by $\mathbb{R}_{\geq 0}^X$, and $\mathbf{0}$ the valuation that maps every clock in X to 0. For $\delta \in \mathbb{R}_{\geq 0}$, we denote $\nu + \delta$ the valuation mapping each $x \in X$ to the value $\nu(x) + \delta$. For a subset R of X , let $[R]\nu$ be the valuation that sets x to 0 if $x \in R$ and assigns $\nu(x)$ otherwise. A *clock constraint* is a conjunction of constraints $x \# c$ for $x \in X$, $\# \in \{<, \leq, =, \geq, >\}$ and $c \in \mathbb{N}$, e.g. We denote $\Phi(X)$ the set of clock constraints over clock variables X . For a valuation ν and a constraint ϕ we write $\nu \models \phi$ when ν satisfies ϕ , that is, when ϕ holds after replacing every x by $\nu(x)$.

A *Timed Automaton (TA)* [1] \mathcal{A} is a finite automaton extended with clocks that enable or disable transitions. Formally, \mathcal{A} is a tuple (Q, q_0, X, T) where Q is a finite set of states, $q_0 \in Q$ is the initial state, X is a finite set of clocks and $T \subseteq Q \times \Phi(X) \times 2^X \times Q$ is a finite set of transitions. For each transition $(q, g, R, q') \in T$, g is a guard that defines the valuations of the clocks that allow to cross the transition, and R is a set of clocks that are reset on the transition.

A configuration of \mathcal{A} is a pair $(q, \nu) \in Q \times \mathbb{R}_{\geq 0}^X$. A transition $(q, \nu) \xrightarrow{\delta, t} (q', \nu')$ with $t = (q, g, R, q') \in T$ and $\delta \in \mathbb{R}_{\geq 0}$ is enabled when $\nu + \delta \models g$ and $\nu' = [R](\nu + \delta)$. A *run* ρ of \mathcal{A} is a (finite or infinite) sequence of transitions starting from the initial configuration $(q_0, \mathbf{0})$: $(q_0, \mathbf{0}) \xrightarrow{\delta_0, t_0} (q_1, \nu_1) \xrightarrow{\delta_1, t_1} \dots (q_i, \nu_i) \xrightarrow{\delta_i, t_i} \dots$

Definition 1 (Zeno/non-Zeno runs). A run $(q_0, \mathbf{0}) \xrightarrow{\delta_0, t_0} \dots (q_i, \nu_i) \xrightarrow{\delta_i, t_i} \dots$ is non-Zeno if time diverges, that is, $\sum_{i \geq 0} \delta_i = \infty$. Otherwise it is Zeno.

Notice that only infinite sequences can be non-Zeno. As can be seen, the number of configurations (q, ν) could be uncountable. We now define the abstract semantics for timed automata.

2.2 Symbolic Semantics, Zenoness and non-Zenoness Problems

A *zone* is a set of clock valuations that satisfy a conjunction of constraints of the form $x_i \# c$ and $x_i - x_j \# c$ with $x_i, x_j \in X$, $\# \in \{<, \leq, =, \geq, >\}$ and $c \in \mathbb{N}$. For instance, $(x_1 \leq 1 \wedge x_1 - x_2 \geq 0)$ is a zone. Zones can be efficiently represented by

Difference Bound Matrices (DBMs) [7]. A DBM representation of a zone Z is a $|X| + 1$ square matrix $(Z_{ij})_{i,j \in [0;|X|]}$ where each entry $Z_{ij} = (c_{i,j}, \preccurlyeq_{ij})$ represents the constraint $x_i - x_j \preccurlyeq_{ij} c_{ij}$ for $c_{ij} \in \mathbb{Z} \cup \{\infty\}$ and $\preccurlyeq_{ij} \in \{<, \leq\}$. The special clock x_0 encodes the value 0.

The *symbolic semantics* (or *zone graph*) of \mathcal{A} is the transition system $ZG(\mathcal{A}) = (S, s_0, \Rightarrow)$ where S is the set of nodes (q, Z) with q a state of \mathcal{A} and Z a zone; $s_0 = (q_0, Z_0)$ with $Z_0 = \{\mathbf{0} + \delta \mid \delta \in \mathbb{R}_{\geq 0}\}$ as the initial node. There exists a transition $(q, Z) \xrightarrow{t} (q', Z')$ with $t = (q, g, R, q') \in T$ if Z' is the set of valuations $[R]\nu + \delta$ for some $\delta \in \mathbb{R}_{\geq 0}$ and some valuation $\nu \in Z$ such that $\nu \models g$. If Z is a zone, then Z' is a zone. Moreover, a DBM representation of Z' can be computed from the DBM representation of Z (see for instance [4]).

However $ZG(\mathcal{A})$ may still be infinite. Several abstractions have been introduced to obtain a finite graph from $ZG(\mathcal{A})$. A *finite abstraction* \mathbf{a} is a map from $\mathcal{P}(\mathbb{R}_{\geq 0}^X)$ to $\mathcal{P}(\mathbb{R}_{\geq 0}^X)$ such that for every zone Z : $\mathbf{a}(Z)$ is a zone, $Z \subseteq \mathbf{a}(Z)$, $\mathbf{a}(\mathbf{a}(Z)) = \mathbf{a}(Z)$ and \mathbf{a} has a finite range. In particular Extra_M [4], Extra_M^+ , Extra_{LU} and Extra_{LU}^+ [2] are well-known finite abstractions. The last two abstractions are usually preferred as they are coarser and hence lead to more efficient algorithms. We define these abstractions below.

Let $L : X \mapsto \mathbb{N} \cup \{-\infty\}$ and $U : X \mapsto \mathbb{N} \cup \{-\infty\}$ be two maps that associate to each clock in \mathcal{A} its maximal lower bound and its maximal upper bound respectively: that is, for every $x \in X$, $L(x)$ is the maximal integer c such that $x > c$ or $x \geq c$ appears in some guard of \mathcal{A} . We let $L(x) = -\infty$ if no such c exists. Similarly, we define $U(x)$ with respect to clock constraints like $x \leq c$ and $x < c$. We define $\text{Extra}_{LU}(Z) = Z^{LU}$ and $\text{Extra}_{LU}^+(Z) = Z^{LU+}$ as:

$$Z_{ij}^{LU} = \begin{cases} (\infty, <) & \text{if } c_{ij} > L(x_i) \\ (-U(x_j), <) & \text{if } -c_{ij} > U(x_j) \\ Z_{ij} & \text{otherwise} \end{cases} \quad \left| \quad Z_{ij}^{LU+} = \begin{cases} (\infty, <) & \text{if } c_{ij} > L(x_i) \\ (\infty, <) & \text{if } -c_{0i} > L(x_i) \\ (\infty, <) & \text{if } -c_{0j} > U(x_j), i \neq 0 \\ (-U(x_j), <) & \text{if } -c_{0j} > U(x_j), i = 0 \\ Z_{ij} & \text{otherwise} \end{cases}$$

where $L(x_0) = U(x_0) = 0$ for the special clock x_0 . The abstraction Extra_M is defined in a similar way than Extra_{LU} by replacing every occurrence of L and U by M which maps every clock x to $\max(L(x), U(x))$. The following property is later used to extend our results for Extra_{LU} to Extra_{LU}^+ .

Theorem 1 ([2]). *For each zone Z , we have: $Z \subseteq \text{Extra}_M(Z) \subseteq \text{Extra}_M^+(Z)$; $Z \subseteq \text{Extra}_{LU}(Z) \subseteq \text{Extra}_{LU}^+(Z)$ and $\text{Extra}_M^+(Z) \subseteq \text{Extra}_{LU}^+(Z)$.*

For two nodes (q, Z) and (q', Z') , we define the relation $(q, Z) \xRightarrow{\mathbf{a}} (q', Z')$ if $(q, Z) \xrightarrow{t} (q', Z'')$ in $ZG(\mathcal{A})$, $Z = \mathbf{a}(Z)$ and $Z' = \mathbf{a}(Z'')$. The *abstract symbolic semantics* (or the *abstract zone graph*) of \mathcal{A} is the transition system $ZG^{\mathbf{a}}(\mathcal{A})$ induced by $\xRightarrow{\mathbf{a}}$ with the initial node $(q_0, \mathbf{a}(Z_0))$, where (q_0, Z_0) is the initial node of $ZG(\mathcal{A})$. We denote by $ZG^{LU}(\mathcal{A})$ the abstract symbolic semantics when abstraction Extra_{LU} is considered, and $ZG^M(\mathcal{A})$ when the abstraction \mathbf{a} is Extra_M .

A *path* in $ZG^{\mathbf{a}}(\mathcal{A})$ is a (finite or infinite) sequence of transitions:

$$(q_0, Z_0) \xRightarrow{t_0}_{\mathbf{a}} (q_1, Z_1) \xRightarrow{t_1}_{\mathbf{a}} \cdots (q_i, Z_i) \xRightarrow{t_i}_{\mathbf{a}} \cdots$$

We say that a run $(q_0, \mathbf{0}) \xrightarrow{\delta_0, t_0} \dots (q_i, \nu_i) \xrightarrow{\delta_i, t_i} \dots$ of \mathcal{A} is an *instance* of a path π of $ZG^{\mathbf{a}}(\mathcal{A})$ if they agree on the sequence of transitions t_0, t_1, \dots , and if for every $i \geq 0$, (q_i, ν_i) and (q_i, Z_i) coincide on q_i , and $\nu_i \in Z_i$. By definition of Z_i this implies $\nu_i + \delta_i \in Z_i$. We say that an abstraction \mathbf{a} is *sound* if every path π can be instantiated as a run of \mathcal{A} . Conversely, \mathbf{a} is *complete* when every run of \mathcal{A} is an instance of some path in $ZG^{\mathbf{a}}(\mathcal{A})$.

A classical verification problem for Timed Automata is to answer state reachability queries. For that purpose, runs of \mathcal{A} and paths in $ZG^{\mathbf{a}}(\mathcal{A})$ are defined as finite sequences of transitions. A reachability query asks for the existence of a finite run leading to a given state. Reachability problems can be solved using $ZG^{\mathbf{a}}(\mathcal{A})$ when \mathbf{a} is sound and complete and this property holds for the classical abstractions.

Theorem 2 ([4, 2]). $\text{Extra}_M, \text{Extra}_M^+, \text{Extra}_{LU}$ and Extra_{LU}^+ are sound and complete for finite sequences of transitions.

Liveness properties ask for the existence of an infinite run satisfying a given property. For instance, does \mathcal{A} visit state q infinitely often? Soundness and completeness of \mathbf{a} with respect to infinite runs allow to solve such problems from $ZG^{\mathbf{a}}(\mathcal{A})$. Recently, it has also been proved that classical abstractions are also sound and complete for infinite paths/runs.

Theorem 3 ([12, 11]). $\text{Extra}_M, \text{Extra}_M^+, \text{Extra}_{LU}$ and Extra_{LU}^+ are sound and complete for infinite sequences of transitions.

Thanks to Theorem 3, we know that every path π in $ZG^{\mathbf{a}}(\mathcal{A})$ can be instantiated to a run of \mathcal{A} . However, soundness is not sufficient to know if π can be instantiated as a *non-Zeno* run. In the sequel, we consider the following problems, given an automaton \mathcal{A} and an abstract zone graph $ZG^{\mathbf{a}}(\mathcal{A})$.

INPUT	\mathcal{A} and $ZG^{\mathbf{a}}(\mathcal{A})$
NON-ZENONESS PROBLEM ($\text{NZP}^{\mathbf{a}}$)	Does \mathcal{A} have a non-Zeno run?
ZENONESS PROBLEM ($\text{ZP}^{\mathbf{a}}$)	Does \mathcal{A} have a Zeno run?

Observe that solving $\text{ZP}^{\mathbf{a}}$ does not solve $\text{NZP}^{\mathbf{a}}$ and vice-versa: one is not the negation of the other. In this paper, we focus on the complexity of deciding $\text{ZP}^{\mathbf{a}}$ and $\text{NZP}^{\mathbf{a}}$ for different abstractions \mathbf{a} . We denote NZP^M and ZP^M when abstraction Extra_M is considered. We similarly define NZP^{LU} and ZP^{LU} for abstraction Extra_{LU} . The non-Zenoness problem is solved in polynomial time when abstraction Extra_M is considered [9, 10]. Surprisingly, this is not true for abstraction Extra_{LU} : in Section 3 we show that NZP^{LU} is NP-complete. The same asymmetry appears in the Zenoness problem as well, which is shown in Section 5.

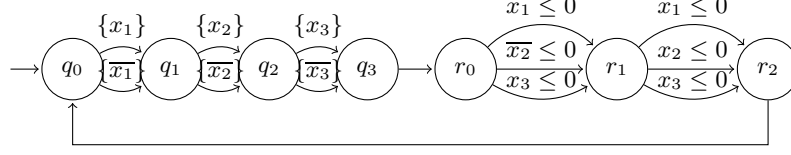


Fig. 1. \mathcal{A}_ϕ^{NZ} for $\phi = (p_1 \vee \neg p_2 \vee p_3) \wedge (\neg p_1 \vee p_2 \vee p_3)$

3 Non-Zenoness is NP-complete for \mathbf{Extra}_{LU}

We give a reduction from the 3SAT problem: given a 3CNF formula ϕ , we build an automaton \mathcal{A}_ϕ^{NZ} that has a non-Zeno run iff ϕ is satisfiable. The size of the automaton will be linear in the size of ϕ . We will then show that the abstract zone graph $ZG^{LU}(\mathcal{A}_\phi^{NZ})$ is isomorphic to the automaton \mathcal{A}_ϕ^{NZ} , thus completing the polynomial reduction from 3SAT to \mathbf{NZP}^{LU} .

Let $P = \{p_1, \dots, p_k\}$ be a set of propositional variables and let $\phi = C_1 \wedge \dots \wedge C_n$ be a 3CNF formula with n clauses. We define the timed automaton \mathcal{A}_ϕ^{NZ} as follows. Its set of clocks X equals $\{x_1, \dots, x_k, \overline{x}_1, \dots, \overline{x}_k\}$. For a literal λ , let $cl(\lambda)$ denote the clock x_i when $\lambda = p_i$ and the clock \overline{x}_i when $\lambda = \neg p_i$. The set of states of \mathcal{A}_ϕ^{NZ} is $\{q_0, \dots, q_k, r_0, \dots, r_n\}$ where q_0 is the initial state. The transitions are as follows:

- for each p_i we have transitions $q_{i-1} \xrightarrow{\{x_i\}} q_i$ and $q_{i-1} \xrightarrow{\{\overline{x}_i\}} q_i$,
- for each clause $C_m = \lambda_1^m \vee \lambda_2^m \vee \lambda_3^m$, $m = 1, \dots, n$, there are three transitions $r_{m-1} \xrightarrow{cl(\lambda_j^m) \leq 0} r_m$ where $\lambda_j^m \in \{\lambda_1^m, \lambda_2^m, \lambda_3^m\}$,
- transitions $q_k \rightarrow r_0$ and $r_n \rightarrow q_0$ with no guards and resets.

Figure 1 shows the automaton for the formula $(p_1 \vee \neg p_2 \vee p_3) \wedge (\neg p_1 \vee p_2 \vee p_3)$. Intuitively, a reset of x_i represents $p_i \mapsto \text{true}$ and a reset of \overline{x}_i means $p_i \mapsto \text{false}$. From r_0 to r_2 we check if the formula is satisfied by this guessed assignment. This formula is satisfied by every assignment that maps p_3 to *true*. This can be seen from the automaton by picking a cycle containing the transitions $q_2 \xrightarrow{\{x_3\}} q_3$, $r_0 \xrightarrow{x_3 \leq 0} r_1$ and $r_1 \xrightarrow{x_3 \leq 0} r_2$. On that path, time can elapse for instance in state q_0 , since x_3 is reset before being zero-checked. Conversely, consider the assignment $p_1 \mapsto \text{false}$, $p_2 \mapsto \text{true}$ and $p_3 \mapsto \text{false}$ that does not satisfy the formula. Take a cycle that resets \overline{x}_1 , x_2 and \overline{x}_3 corresponding to the assignment. Then none of the clocks that are checked for zero on the transitions from r_0 to r_1 has been reset. Notice that these transitions come from the first clause in the formula that evaluates to *false* according to the assignment. To take a transition from r_0 , one of x_1 , \overline{x}_2 and x_3 must be zero and hence time cannot elapse.

Lemma 1 below states that if the formula is satisfiable, there exists a sequence of resets that allows time elapse in every loop. Conversely, if the formula is unsatisfiable, in every iteration of the loop, there is a zero-check that prevents time from elapsing. The proof of Lemma 1 is given in Appendix A.

Lemma 1. *A 3CNF formula ϕ is satisfiable iff \mathcal{A}_ϕ^{NZ} has a non-Zeno run.*

The NP-hardness of NZP^{LU} then follows due to the small size of $ZG^{LU}(\mathcal{A}_\phi^{NZ})$.

Theorem 4. *The abstract zone graph $ZG^{LU}(\mathcal{A}_\phi^{NZ})$ is isomorphic to \mathcal{A}_ϕ^{NZ} . The non-Zenoness problem is NP-complete for abstractions Extra_{LU} and Extra_{LU}^+ .*

Proof. We first prove that $ZG^{LU}(\mathcal{A}_\phi^{NZ})$ is isomorphic to \mathcal{A}_ϕ^{NZ} . For every clock x , $L(x) = -\infty$, hence Extra_{LU} abstracts all the constraints $x_i - x_j \preccurlyeq_{ij} c_{ij}$ to $x_i - x_j < \infty$ except those of the form $x_0 - x_i \preccurlyeq_{0i} c_{0i}$ that are kept unchanged. Due to the guards in \mathcal{A}_ϕ^{NZ} , for every reachable zone in $ZG(\mathcal{A}_\phi^{NZ})$ we have $x_0 - x_i \leq 0$ (i.e. $x_i \geq 0$). Therefore $\text{Extra}_{LU}(Z)$ is the zone defined by $\bigwedge_{x \in X} x \geq 0$ which is $\mathbb{R}_{\geq 0}^X$. For each state of \mathcal{A}_ϕ^{NZ} , the zone $\mathbb{R}_{\geq 0}^X$ is the only reachable zone in $ZG^{LU}(\mathcal{A}_\phi^{NZ})$, hence showing the isomorphism. The result transfers to Extra_{LU}^+ thanks to Theorem 1.

The NP-hardness of NZP^{LU} then follows from Lemma 1. The membership to NP will be proved in Lemma 3 in the next section. \square

Notice that the type of zero checks in \mathcal{A}_ϕ^{NZ} is crucial to Theorem 4. Replacing zero-checks of the form $x \leq 0$ by $x = 0$ does not modify the semantics of \mathcal{A}_ϕ^{NZ} . However, this yields $L(x) = 0$ for every clock x . Hence, the constraints of the form $x_i - x_j \leq 0$ are not abstracted: Extra_{LU} then preserves the ordering among the clocks. Each sequence of clock resets leading from q_0 to q_k yields a distinct ordering on the clocks. Thus, there are exponentially many LU-abstracted zones with state q_k . As a consequence, the polynomial reduction from 3SAT is lost. We indeed provide in Section 4 below an algorithm for detecting non-Zeno runs from $ZG^{LU}(\mathcal{A})$ that runs in polynomial time when $L(x) = 0$ for every clock x .

4 Finding non-Zeno runs

Recall the non-Zenoness problem (NZP^a):

Given an automaton \mathcal{A} and its abstract zone graph $ZG^a(\mathcal{A})$, decide if \mathcal{A} has a non-Zeno run.

A standard solution to this problem involves adding one auxiliary clock to \mathcal{A} to detect non-Zenoness [12]. This solution was shown to cause an exponential blowup in [9]. In the same paper, a polynomial method has been proposed in the case of the Extra_M abstraction. We extend this method to an arbitrary abstraction \mathbf{a} and give conditions on \mathbf{a} for the method to remain polynomial.

An infinite run of the timed automaton could be Zeno due to two factors: *blocking clocks*, which are clocks that are bounded from above (i.e. $x \leq c$ for some $c > 0$) but are never reset in the run and *zero checks*, which are guards of the form $x \leq 0$ or $x = 0$ that prevent time elapse in the run. The method in [9] tackles these two problems as follows. Blocking clocks are handled by first detecting a maximal strongly connected component (SCC) of the zone graph

and repeatedly discarding the transitions that bound some blocking clock until a non-trivial SCC with no such clocks is obtained. This algorithm runs in time polynomial for every abstraction that is sound and complete. For zero checks, a *guessing zone graph* construction has been introduced to detect nodes where time can elapse. We now extend this construction to an arbitrary abstraction.

4.1 Reduced guessing zone graph $rGZG^a(\mathcal{A})$

The necessary and sufficient condition for time elapse in a node despite zero-checks is to have every reachable zero-check from that node preceded by a corresponding reset. The nodes of the guessing zone graph are triples (q, Z, Y) where $Y \subseteq X$ is the set of clocks that can potentially be checked for zero before being reset in a path from (q, Z, Y) . In particular, in a node with $Y = \emptyset$ zero-checks do not hinder time elapse.

A clock that is never checked for zero need not be remembered in sets Y . In order to lift the construction in [9], we restrict Y sets to only contain clocks that can indeed be checked for zero. We say that a clock x is *relevant* if there exists a guard $x \leq 0$ or $x = 0$ in the automaton. We denote the set of relevant clocks by $\text{Rl}(\mathcal{A})$. For a zone Z , let $\mathcal{C}_0(Z)$ denote the set of clocks x such that there exists a valuation $\nu \in Z$ with $\nu(x) = 0$. The clocks that can be checked for zero from (q, Z) lie in $\text{Rl}(\mathcal{A}) \cap \mathcal{C}_0(Z)$.

Definition 2. Let \mathcal{A} be a timed automaton with clocks X . The reduced guessing zone graph $rGZG^a(\mathcal{A})$ has nodes of the form (q, Z, Y) where (q, Z) is a node in $ZG^a(\mathcal{A})$ and $Y \subseteq \text{Rl}(\mathcal{A}) \cap \mathcal{C}_0(Z)$. The initial node is $(q_0, Z_0, \text{Rl}(\mathcal{A}))$, with (q_0, Z_0) the initial node of $ZG^a(\mathcal{A})$. For $t = (q, g, R, q')$, there is a transition $(q, Z, Y) \xrightarrow{t}_a (q', Z', Y')$ with $Y' = (Y \cup R) \cap \text{Rl}(\mathcal{A}) \cap \mathcal{C}_0(Z')$ if there is $(q, Z) \xrightarrow{t}_a (q', Z')$ in $ZG^a(\mathcal{A})$ and some valuation $\nu \in Z$ such that $\nu \models (\text{Rl}(\mathcal{A}) - Y) > 0$ and $\nu \models g$. A new auxiliary letter τ is introduced that adds transitions $(q, Z, Y) \xrightarrow{\tau}_a (q, Z, Y')$ for $Y' = \emptyset$ or $Y' = Y$.

Observe that as we require $\nu \models (\text{Rl}(\mathcal{A}) - Y) > 0$ and $\nu \models g$ for some $\nu \in Z$, a transition that checks $x \leq 0$ (or $x = 0$) is allowed from a node (q, Z, Y) only if $x \in Y$. Thus, from a node (q, Z, \emptyset) every reachable zero-check should be preceded by the corresponding reset. Such a node is called *clear*. Time can elapse in clear nodes. A variable x is *bounded* in a transition of $rGZG^a$ if the guard of the transition implies $x \leq c$ for some constant c . A path of $rGZG^a$ is said to be *blocked* if there is a variable that is bounded infinitely often and reset only finitely often by the transitions on the path. Otherwise the path is called *unblocked*. An unblocked path says that there are no blocking clocks to bound time and clear nodes suggest that inspite of zero-checks that might possibly occur in the future, time can still elapse. We get the following theorem.

Theorem 5. A timed automaton \mathcal{A} has a non-Zeno run iff there exists an unblocked path in $rGZG^a(\mathcal{A})$ visiting a clear node infinitely often.

The proof of Theorem 5 follows from Lemmas 11 and 12 in Appendix B. The proof is in the same lines as for the guessing zone graph in [9].

4.2 Polynomial algorithms for NZP^a

Since we have a node in $rGZG^a(\mathcal{A})$ for every (q, Z) in $ZG^a(\mathcal{A})$ and every subset Y of $\text{Rl}(\mathcal{A})$, it can in principle be exponentially bigger than $ZG^a(\mathcal{A})$. Below, we see that depending on abstraction \mathfrak{a} , not all subsets Y need to be considered.

Let X' be a subset of X . We say that a zone Z *orders the clocks in X'* if for all clocks $x, y \in X'$, Z implies that at least one of $x \leq y$ or $y \leq x$ hold.

Definition 3 (Weakly order-preserving abstractions). *An abstraction \mathfrak{a} weakly preserves orders if for all clocks $x, y \in \text{Rl}(\mathcal{A}) \cap \mathcal{C}_0(Z)$, $Z \models x \leq y$ iff $\mathfrak{a}(Z) \models x \leq y$.*

It has been observed in [9] that all the zones that are reachable in the unabstracted zone graph $ZG(\mathcal{A})$ order the entire set of clocks X . Assume that \mathfrak{a} weakly preserves orders, then for every reachable node (q, Z, Y) in $rGZG^a(\mathcal{A})$, the zone Z orders the clocks in $\text{Rl}(\mathcal{A}) \cap \mathcal{C}_0(Z)$. We now show that Y is downward closed with respect to this order given by Z : for clocks $x, y \in \text{Rl}(\mathcal{A}) \cap \mathcal{C}_0(Z)$, if $Z \models x \leq y$ and $y \in Y$, then $x \in Y$. This entails that there are at most $\text{Rl}(\mathcal{A})$ downward closed sets to consider, thus giving a polynomial complexity.

Theorem 6. *Let \mathcal{A} be a timed automaton. If \mathfrak{a} weakly preserves orders, then the reachable part of $rGZG^a(\mathcal{A})$ is $\mathcal{O}(|\text{Rl}(\mathcal{A})|)$ bigger than the reachable part of $ZG^a(\mathcal{A})$.*

Proof. We prove by induction on the transitions in $rGZG^a(\mathcal{A})$ that for every reachable node (q, Z, Y) the set Y is downward closed with respect to Z on the clocks in $\text{Rl}(\mathcal{A}) \cap \mathcal{C}_0(Z)$. This is true for the initial node $(q_0, Z_0, \text{Rl}(\mathcal{A}))$.

Now, assume that this is true for (q, Z, Y) . Take a transition $(q, Z, Y) \xrightarrow{t}_a (q', Z', Y')$ with $t = (q, g, R, q')$. By definition, $Y' = (Y \cup R) \cap \text{Rl}(\mathcal{A}) \cap \mathcal{C}_0(Z')$. Suppose $Z' \models x \leq y$ for some $x, y \in \text{Rl}(\mathcal{A}) \cap \mathcal{C}_0(Z')$ and suppose $y \in Y'$. This could mean $y \in Y$ or $y \in R$. If $y \in R$, then x is also in R since $Z' \models x \leq y$. If $y \notin R$ then we get $y \in Y$ and $Z \models x \leq y$. By hypothesis that Y is downward closed, $x \in Y$. In both cases $x \in Y'$. \square

The definition of Extra_M in section 2.2 clearly shows that it weakly preserves orders. Hence, $rGZG^M(\mathcal{A})$ yields a polynomial algorithm for NZP^M . Notice that thanks to the reduction of the guessing zone graph to the relevant clocks, we propose an algorithm that is more efficient than the algorithm in [9] despite using the same abstraction.

Lemma 2. *The abstractions Extra_M , Extra_M^+ weakly preserve orders.*

Proof. It has been proved in [9] that Extra_M weakly preserves orders. Note that for a clock x in $\text{Rl}(\mathcal{A})$ we have $M(x) \geq 0$ and so if $x \in \text{Rl}(\mathcal{A}) \cap \mathcal{C}_0(Z)$, then it means that Z is consistent with $x \leq M(x)$. Therefore, by definition, $\text{Extra}_M^+(Z)$ restricted to clocks in $\text{Rl}(\mathcal{A}) \cap \mathcal{C}_0(Z)$ is identical to $\text{Extra}_M(Z)$ restricted to the same set of clocks. Since Extra_M is weakly order preserving, we get that Extra_M^+ is weakly order preserving too. \square

However, the polynomial complexity is not preserved by coarser abstractions Extra_{LU} and Extra_{LU}^+ .

Lemma 3. *The abstractions Extra_{LU} and Extra_{LU}^+ do not weakly preserve orders. The non-Zenoness problem is in NP for Extra_{LU} and Extra_{LU}^+ .*

Proof. The proof of Theorem 4 gives an example that illustrates Extra_{LU} does not weakly preserve orders. This also holds for Extra_{LU}^+ by Theorem 1.

For the NP membership, let N be the number of nodes in $ZG^{LU}(\mathcal{A})$. Let us non-deterministically choose a node (q, Z) . We assume that (q, Z) is reachable as this can be checked in polynomial time on $ZG^{LU}(\mathcal{A})$.

We augment (q, Z) with an empty guess set of clocks. From (q, Z, \emptyset) , we non-deterministically simulate a path π of the (non-reduced) guessing zone graph [9] obtained from Definition 2 with $\text{Rl}(\mathcal{A}) = X$ and $\mathcal{C}_0(Z) = X$ for every zone Z . We avoid taking τ transitions on this path. This ensures that the guess sets accumulate all the resets on π . During the simulation, we also keep track of a separate set U containing all the clocks that are bounded from above on a transition in π .

If during the simulation one reaches a node (q, Z, Y) such that $U \subseteq Y$, then we have a cycle $(q, Z, \emptyset) \Rightarrow_a^* (q, Z, Y) \xrightarrow{\tau}_a (q, Z, \emptyset)$ that is unblocked and that visits a clear node infinitely often. Also, since (q, Z) is reachable in $ZG^{LU}(\mathcal{A})$, (q, Z, X) is reachable in the guessing zone graph. Then (q, Z, \emptyset) is reachable from (q, Z, X) with a τ transition. From [9] and from the fact that Extra_{LU} and Extra_{LU}^+ are sound and complete [2] we get a non-Zeno run of \mathcal{A} .

Notice that it is sufficient to simulate $N \times (|X| + 1)$ transitions since we can avoid visiting a node (q', Z', Y') twice in π . \square

The abstraction Extra_{LU} does not weakly preserve order in zones due to relevant clocks with $L(x) = -\infty$ and $U(x) \geq 0$. We show that this is the only reason for NP-hardness. We slightly modify Extra_{LU} to get an abstraction $\text{Extra}_{\overline{L}U}$ that is coarser than Extra_M , but it still weakly preserves orders.

Definition 4 (Weak L bounds). *Let \mathcal{A} be a timed automaton. Given the bounds $L(x)$ and $U(x)$ for every clock $x \in X$, the weak lower bound \overline{L} is given by: $\overline{L}(x) = 0$ if $x \in \text{Rl}(\mathcal{A})$, $L(x) = -\infty$ and $U(x) \geq 0$, and $\overline{L}(x) = L(x)$ otherwise.*

We denote $\text{Extra}_{\overline{L}U}$ the Extra_{LU} abstraction obtained by choosing \overline{L} instead of L . Notice that $\text{Extra}_{\overline{L}U}$ and Extra_{LU} coincide when zero-checks are written $x = 0$ instead of $x \leq 0$ in the automaton. By definition of Extra_{LU} , we get the following.

Lemma 4. *The abstraction $\text{Extra}_{\overline{L}U}$ weakly preserves orders.*

$\text{Extra}_{\overline{L}U}$ coincides with Extra_{LU} for a wide class of automata. For instance, when the automaton does not have a zero-check, $\text{Extra}_{\overline{L}U}$ is exactly Extra_{LU} , and the existence of a non-Zeno run can be decided in polynomial time.

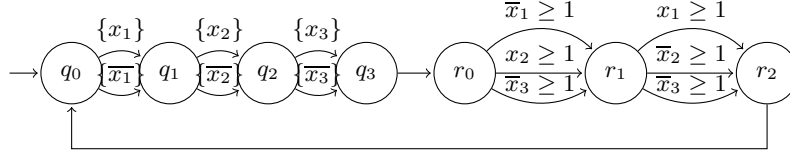


Fig. 2. \mathcal{A}_ϕ^Z for $\phi = (p_1 \vee \neg p_2 \vee p_3) \wedge (\neg p_1 \vee p_2 \vee p_3)$

5 The Zenoness problem

In this section we consider the Zenoness problem (ZP^a):

Given an automaton \mathcal{A} and its abstract zone graph $ZG^a(\mathcal{A})$, decide if \mathcal{A} has a Zeno run.

As in the case of non-Zenoness, this problem turns out to be NP-complete when the abstraction operator \mathbf{a} is Extra_{LU} . We subsequently give the hardness proof by providing a reduction from 3SAT.

5.1 Reducing 3SAT to ZP^a with abstraction Extra_{LU}

Let $P = \{p_1, \dots, p_k\}$ be a set of propositional variables. Let $\phi = C_1 \wedge \dots \wedge C_n$ be a 3CNF formula with n clauses. Each clause C_m , $m = 1, 2, \dots, n$ is a disjunction of three literals λ_1^m , λ_2^m and λ_3^m . We construct in polynomial time an automaton \mathcal{A}_ϕ^Z and its zone graph $ZG^{LU}(\mathcal{A}_\phi^Z)$ such that \mathcal{A}_ϕ^Z has a Zeno run iff ϕ is satisfiable, thus proving the NP-hardness.

The automaton \mathcal{A}_ϕ^Z has clocks $\{x_1, \overline{x}_1, \dots, x_k, \overline{x}_k\}$ with x_i and \overline{x}_i corresponding to the literals p_i and $\neg p_i$ respectively. We denote the clock associated to a literal λ by $cl(\lambda)$. The set of states of \mathcal{A}_ϕ^Z is given by $\{q_0, q_1, \dots, q_k\} \cup \{r_0, r_1, r_2, \dots, r_n\}$ with q_0 being the initial state. The transitions are as follows:

- transitions $q_{i-1} \xrightarrow{\{x_i\}} q_i$ and $q_{i-1} \xrightarrow{\{\overline{x}_i\}} q_i$ for $i = 1, 2, \dots, k$,
- a transition $q_k \rightarrow r_0$ with no guards and resets,
- for each clause C_m there are three transitions $r_{m-1} \xrightarrow{cl(\overline{\lambda}) \geq 1} r_m$ where $\lambda = \{\lambda_1^m, \lambda_2^m, \lambda_3^m\}$,
- a transition $r_n \rightarrow q_0$ with no guards and resets. This transition creates a cycle in \mathcal{A}_ϕ^Z .

As an example, Figure 2 shows the automaton for the formula $(p_1 \vee \neg p_2 \vee p_3) \wedge (\neg p_1 \vee p_2 \vee p_3)$. Clearly, the automaton \mathcal{A}_ϕ^Z can be constructed from ϕ in $\mathcal{O}(n)$ time. It remains to show that $ZG^{LU}(\mathcal{A}_\phi^Z)$ can also be calculated in polynomial time from \mathcal{A}_ϕ^Z and to show that ϕ is satisfiable iff \mathcal{A}_ϕ^Z has a Zeno run. This is proved below.

Lemma 5. *A 3CNF formula ϕ is satisfiable iff \mathcal{A}_ϕ^Z has a Zeno run.*

The proof of Lemma 5 is given in Appendix C. We note that the size of the $ZG^{LU}(\mathcal{A})$ is the same as that of the automaton.

Theorem 7. *The zone graph $ZG^{LU}(\mathcal{A}_\phi^Z)$ is isomorphic to \mathcal{A}_ϕ^Z . The Zenoness problem is NP-complete for Extra_{LU} and Extra_{LU}^+ .*

Proof. By looking at the guards in the transitions, we get that for each clock x , $L(x) = 1$ and $U(x) = -\infty$. The initial node of the zone graph $ZG^{LU}(\mathcal{A}_\phi^Z)$ is $(q_0, \text{Extra}_{LU}(Z_0))$ where Z_0 is the set of valuations given by $(x_1 \geq 0) \wedge (x_1 = \overline{x_1} = \dots = x_k = \overline{x_k})$. By definition, since for each clock x , $U(x) = -\infty$, we have $\text{Extra}_{LU}(Z_0) = \mathbb{R}_{\geq 0}^X$, the non-negative half-space.

After resetting a clock x in a transition from $\mathbb{R}_{\geq 0}^X$, we get back to $\mathbb{R}_{\geq 0}^X$. On taking a transition with a guard $x \geq 1$ from $\mathbb{R}_{\geq 0}^X$, we come to a zone $\mathbb{R}_{\geq 0}^X \wedge x \geq 1$. However, since $U(x) = -\infty$, $\text{Extra}_{LU}(\mathbb{R}_{\geq 0}^X \wedge x \geq 1)$ gives back $\mathbb{R}_{\geq 0}^X$. It follows that $ZG^{LU}(\mathcal{A}_\phi^Z)$ is isomorphic to \mathcal{A}_ϕ^Z . This extends to Extra_{LU}^+ by Theorem 1.

NP-hardness then comes from Lemma 5. NP-membership is proved in Lemma 7. \square

In the next section, we provide an algorithm for the zenoness problem ZP^a and give conditions on abstraction \mathbf{a} for the solution to be polynomial.

5.2 Finding Zeno paths

We say that a transition is *lifting* if it has a guard that implies $x \geq 1$ for some clock x . The idea is to find if there exists a run of an automaton \mathcal{A} in which every clock x that is reset infinitely often is lifted only finitely many times, ensuring that the run is Zeno. This amounts to checking if there exists a cycle in $ZG(\mathcal{A})$ where every clock that is reset is not lifted. Observe that when $(q, Z) \xrightarrow{x \geq c} (q', Z')$ is a transition of $ZG(\mathcal{A})$, then Z' entails that $x \geq c$. Therefore, if a node (q, Z) is part of a cycle in the required form, then in particular, all the clocks that are greater than 1 in Z should not be reset in the cycle.

Based on the above intuition, our solution begins with computing the zone graph on-the-fly. At some node (q, Z) the algorithm non-deterministically guesses that this node is part of a cycle that yields a zeno run. This node transits to what we call the *slow mode*. In this mode, a reset of x in a transition is allowed from (q', Z') only if Z' is consistent with $x < 1$.

Before we define our construction formally, recall that we would be working with the abstract zone graph $ZG^a(\mathcal{A})$ and not $ZG(\mathcal{A})$. Therefore for our solution to work, the abstraction operator \mathbf{a} should remember the fact that a clock has a value greater than 1.

For an automaton \mathcal{A} over the set of clocks X , let $\text{Lf}(\mathcal{A})$ denote the set of clocks appear in a lifting transition of \mathcal{A} .

Definition 5 (Lift-safe abstractions). *An abstraction operator \mathbf{a} is called lift-safe if for every zone Z and for every clock $x \in \text{Lf}(\mathcal{A})$, if $Z \models x \geq 1$ then $\mathbf{a}(Z) \models x \geq 1$.*

We are now in a position to define our *slow zone graph* construction to decide if an automaton has a Zeno run.

Definition 6 (Slow zone graph). Let \mathcal{A} be a timed automaton over the set of clocks X . Let \mathbf{a} be a lift-safe abstraction. The slow zone graph $SZG^{\mathbf{a}}(\mathcal{A})$ has nodes of the form (q, Z, l) where $l = \{\text{free}, \text{slow}\}$. The initial node is (q_0, Z_0, free) where (q_0, Z_0) is the initial node of $ZG^{\mathbf{a}}(\mathcal{A})$. For every transition $(q, Z) \xrightarrow{t}_{\mathbf{a}} (q', Z')$ in $ZG^{\mathbf{a}}(\mathcal{A})$ with $t = (g, R, q')$, we have the following transitions in $SZG^{\mathbf{a}}(\mathcal{A})$:

- a transition $(q, Z, \text{free}) \xrightarrow{t}_{\mathbf{a}} (q', Z', \text{free})$,
- a transition $(q, Z, \text{slow}) \xrightarrow{t}_{\mathbf{a}} (q', Z', \text{slow})$ if for all clocks $x \in R$, $Z \wedge g$ is consistent with $x < 1$,

A new letter τ is introduced that adds transitions $(q, Z, \text{free}) \xrightarrow{\tau}_{\mathbf{a}} (q, Z, \text{slow})$.

A node of the form (q, Z, slow) is said to be a *slow node*. A path of $SZG^{\mathbf{a}}(\mathcal{A})$ is said to be *slow* if it has a suffix consisting entirely of slow nodes. The τ -transitions take a node (q, Z) from the *free* mode to the *slow* mode. Note that the transitions of the slow mode are constrained further. Lemmas 13 and 14 in Appendix D show that there is a cycle in the $SZG^{\mathbf{a}}(\mathcal{A})$ consisting entirely of slow nodes iff \mathcal{A} has a Zeno run.

The above two lemmas prove the correctness of the approach. From the definition of $SZG^{\mathbf{a}}(\mathcal{A})$ it follows clearly that for each node (q, Z) of the zone graph there are two nodes in $SZG^{\mathbf{a}}(\mathcal{A})$: (q, Z, free) and (q, Z, slow) . We thus get the following theorem.

Theorem 8. Let \mathbf{a} be a lift-safe abstraction. The automaton \mathcal{A} has a Zeno run iff $SZG^{\mathbf{a}}(\mathcal{A})$ has an infinite slow path. The number of reachable nodes of $SZG^{\mathbf{a}}(\mathcal{A})$ is at most twice the number of reachable nodes in $ZG^{\mathbf{a}}(\mathcal{A})$.

We now turn our attention towards some of the abstractions existing in the literature. We observe that the classical Extra_M is lift-safe and hence the Zenoness problem could be solved using the slow zone graph construction. However, in accordance to the NP-hardness of the problem for Extra_{LU} , we get that Extra_{LU} is not lift-safe.

Lemma 6. The abstractions Extra_M , Extra_M^+ are lift-safe.

Proof. Observe that for every clock that is lifted, the bound M is at least 1. It is now straightforward from the definitions of Extra_M , Extra_M^+ that they are lift-safe. \square

Lemma 7. The abstractions Extra_{LU} and Extra_{LU}^+ are not lift-safe. The Zenoness problem for Extra_{LU} and Extra_{LU}^+ is in NP.

Proof. That Extra_{LU} and Extra_{LU}^+ are not lift-safe follows from the proof of Theorem 7. We show the NP-membership using a technique similar to the slow zone graph construction. Since Extra_{LU} is not lift-safe, the reachable zones in $ZG^{LU}(\mathcal{A})$ do not maintain the information about the clocks that have been lifted. Therefore, at some reachable zone (q, Z) we non-deterministically guess the set of clocks Y that are allowed to be lifted in the future and go to a node (q, Z, Y) . From now on, there are transitions $(q, Z, Y) \xrightarrow{t}_a (q', Z', Y)$ when:

- $(q, Z) \xrightarrow{t}_a (q', Z')$ is a transition in $ZG^{LU}(\mathcal{A})$,
- if t contains a guard $x \geq c$ with $c \geq 1$, then $x \in Y$,
- if t resets a clock x , then $x \notin Y$

If a cycle is obtained that contains (q, Z, Y) , then the clocks that are reset and lifted in this cycle are disjoint and hence \mathcal{A} has a Zeno run.

This shows that if \mathcal{A} has a Zeno run we can non-deterministically choose a path of the above form and the length of this path is bounded by twice the number of zones in $ZG^{LU}(\mathcal{A})$ (which is our other input). This proves the NP-membership. \square

5.3 Weakening the U bounds

We saw in Lemma 7 that the extrapolation Extra_{LU} is not lift-safe. This is due to clocks x that are lifted but have $U(x) = -\infty$. These are exactly the clocks x with $L(x) \geq 1$ and $U(x) = -\infty$. We propose to weaken the U bounds so that the information about a clock being lifted is remembered in the abstracted zone.

Definition 7 (Weak U bounds). *Given the bounds $L(x)$ and $U(x)$ for each clock $x \in X$, the weak upper bound $\overline{U}(x)$ is given by: $\overline{U}(x) = 1$ if $L(x) \geq 1$ and $U(x) = -\infty$, and $\overline{U}(x) = U(x)$ otherwise.*

Let $\text{Extra}_{L\overline{U}}$ denote the Extra_{LU} abstraction, but with \overline{U} bound for each clock instead of U . This definition ensures that for all lifted clocks, that is, for all $x \in \text{Lf}(\mathcal{A})$, if a zone entails that $x \geq 1$ then $\text{Extra}_{L\overline{U}}(Z)$ also entails that $x \geq 1$. This is summarized by the following lemma, the proof of which follows by definitions.

Lemma 8. *For all zones Z , $\text{Extra}_{L\overline{U}}$ is lift-safe.*

From Theorem 8, we get that the Zenoness problem is polynomial for $\text{Extra}_{L\overline{U}}$. However, there is a price to pay. Weakening the U bounds leads to zone graphs exponentially bigger in some cases. For example, for the automaton \mathcal{A}_ϕ^Z that was used to prove the NP-completeness of the Zenoness problem with Extra_{LU} , note that the zone graph $ZG^{L\overline{U}}(\mathcal{A}_\phi^Z)$ obtained by applying $\text{Extra}_{L\overline{U}}$ is exponentially bigger than $ZG^{LU}(\mathcal{A}_\phi^Z)$. This leads to a slow zone graph $SZG^{L\overline{U}}(\mathcal{A}_\phi^Z)$ with size polynomial in $ZG^{L\overline{U}}(\mathcal{A}_\phi^Z)$.

6 Conclusion

We have shown a surprising fact that the problem of deciding existence of Zeno or non-Zeno behaviours from abstract zone graphs depends heavily on the abstractions, to the extent that the problem changes from being polynomial to becoming NP-complete as the abstractions get coarser. We have proved NP-completeness for the coarse abstractions Extra_{LU} and Extra_{LU}^+ . In contrast, the fundamental notions of reachability and Büchi emptiness over abstract zone graphs have a mere linear complexity, independent of the abstraction.

On the positive side, from our study on the conditions for an abstraction to give a polynomial solution, we see that a small modification of the LU-extrapolation works. We have defined two weaker abstractions: $\text{Extra}_{\overline{LU}}$ for detecting non-Zeno runs and $\text{Extra}_{\overline{LU}}^+$ for detecting Zeno runs. The weak bounds \overline{L} and \overline{U} can also be used with Extra_{LU}^+ to achieve similar results. Despite leading to a polynomial solution for checking Zeno or non-Zeno behaviours from abstract zone graphs, these abstractions transfer the complexity to the input: they could lead to exponentially bigger abstract zone graphs themselves.

While working with abstract zone graphs, coarse abstractions (and hence small abstract zone graphs) are essential to handle big models of timed automata. These, as we have seen, work against the Zenoness questions. Our results therefore provide a theoretical motivation to look for cheaper substitutes to the notion of Zenoness.

Acknowledgements. We would like to thank Igor Walukiewicz for his insightful ideas on our problems and his helpful comments during the preparation of the manuscript. We thank Laurent Fribourg for pointing us to the problem of detecting Zeno runs.

References

1. R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
2. G. Behrmann, P. Bouyer, K. G. Larsen, and R. Pelanek. Lower and upper bounds in zone-based abstractions of timed automata. *Int. Journal on Software Tools for Technology Transfer*, 8(3):204–215, 2006.
3. G. Behrmann, A. David, K.G. Larsen, J. Haakansson, P. Pettersson, W. Yi, and M. Hendriks. Uppaal 4.0. In *QEST*, pages 125–126. IEEE Computer Society, 2006.
4. P. Bouyer. Forward analysis of updatable timed automata. *Formal Methods in System Design*, 24(3):281–320, 2004.
5. H. Bowman and R. Gómez. How to stop time stopping. *Formal aspects of computing*, 18(4):459–493, 2006.
6. M. Bozga, C. Daws, O. Maler, A. Olivero, S. Tripakis, and S. Yovine. Kronos: A model-checking tool for real-time systems. In *CAV*, volume 1427 of *LNCS*, pages 546–550, 1998.
7. D. Dill. Timing assumptions and verification of finite-state concurrent systems. In *AVMFSS*, volume 407 of *LNCS*, pages 197–212, 1989.

8. R. Gómez and H. Bowman. Efficient detection of zeno runs in timed automata. In *FORMATS*, volume 4763 of *LNCS*, pages 195–210, 2007.
9. F. Herbreteau, B. Srivathsan, and I. Walukiewicz. Efficient emptiness check for timed Büchi automata. In *CAV*, volume 6174 of *LNCS*, pages 148–161, 2010.
10. F. Herbreteau, B. Srivathsan, and I. Walukiewicz. Efficient emptiness check for timed Büchi automata - Extended version. *CoRR abs/1104.1540*, Submitted.
11. G. Li. Checking timed Büchi automata emptiness using lu-abstractions. In *FORMATS*, volume 5813 of *LNCS*, pages 228–242, 2009.
12. S. Tripakis. Checking timed Büchi emptiness on simulation graphs. *ACM Trans. on Computational Logic*, 10(3), 2009.
13. S. Tripakis, S. Yovine, and A. Bouajjani. Checking timed Büchi automata emptiness efficiently. *Formal Methods in System Design*, 26(3):267–292, 2005.
14. Farn Wang. Redlib for the formal verification of embedded systems. In *ISoLA*, pages 341–346. IEEE, 2006.

A Proof of Lemma 1

Lemma 9. *Let ϕ be a satisfiable 3CNF formula, then \mathcal{A}_ϕ^{NZ} has a non-Zeno run*

Proof. Assume that ϕ is satisfied by some variable assignment χ . Let ρ be a sequence of transitions such that:

- from each configuration (q_{i-1}, ν) , with $i \in [1; k]$, ρ instantiates the transition $q_{i-1} \xrightarrow{\{x_i\}} q_i$ when $\chi(p_i) = \text{true}$ and the transition $q_{i-1} \xrightarrow{\{\bar{x}_i\}} q_i$ otherwise;
- from each configuration (r_{m-1}, ν) with $m \in [1; n]$, ρ takes a transition $r_{m-1} \xrightarrow{cl(\lambda_j^m) \leq 0} r_m$ such that λ_j^m evaluates to *true* with respect to χ ;
- and ρ let 1 time unit elapse from each configuration with state r_n .

Now, we prove that ρ is a run of \mathcal{A}_ϕ^{NZ} . We need to prove that zero-checked transitions can be crossed despite elapsing 1 time unit. Recall that every infinite run visits infinitely often a configuration with state r_n . Consider two successive configurations on ρ with state r_n .

$$\dots (r_n, \nu) \xrightarrow{1} \dots \xrightarrow{\{cl(\lambda_j^m)\}} \dots (r_{m-1}, \nu'') \xrightarrow{cl(\lambda_j^m) \leq 0} (r_m, \nu'') \dots (r_n, \nu') \xrightarrow{1} \dots$$

By definition of ρ , λ_j^m is a literal that evaluates to *true* according to χ . Hence, the clock $cl(\lambda_j^m)$ is reset before being zero-checked and $\nu''(cl(\lambda_j^m)) = 0$. As a consequence, the run ρ exists. Furthermore, it is non-Zeno as 1 time unit elapses infinitely often. \square

Lemma 10. *A 3CNF formula ϕ is satisfiable if \mathcal{A}_ϕ^{NZ} has a non-Zeno run*

Proof. Consider a non-Zeno run ρ of \mathcal{A}_ϕ^{NZ} . Since ρ is non-Zeno, time elapses on infinitely many transitions in the run. Every infinite runs of \mathcal{A}_ϕ^{NZ} visits infinitely often a configuration with state r_n . Consider two consecutive configurations on ρ such that time elapses on some transition on the segment from (r_n, ν) to (r_n, ν'') .

$$\dots (r_n, \nu) \rightarrow \dots (q_k, \nu') \rightarrow \dots (r_{m-1}, \nu'') \xrightarrow{cl(\lambda_j^m) \leq 0} (r_m, \nu'') \dots \rightarrow (r_n, \nu'') \dots$$

By construction, for each $i \in [1; k]$ either x_i or \bar{x}_i is reset on the segment from (r_n, ν) to (q_k, ν') . Let χ be the variable assignment that associates *true* to p_i when x_i is reset, and *false* otherwise, that is when \bar{x}_i is reset. We prove that χ satisfies ϕ .

Consider the transition $(r_{m-1}, \nu'') \xrightarrow{cl(\lambda_j^m) \leq 0} (r_m, \nu'')$. It must be the case that $\nu''(cl(\lambda_j^m)) = 0$. Notice that time cannot elapse from (q_k, ν') to (r_n, ν'') because of zero-checks. Hence, time elapse can occur between (r_n, ν) and (q_k, ν') . Thus the clock $cl(\lambda_j^m)$ must be reset before reaching (r_{m-1}, ν'') . Thus, $\chi(\lambda_j^m) = \text{true}$, hence C_m also evaluates to true. This holds for all the clauses. As a consequence, ϕ is satisfied by χ . \square

B Proof of Theorem 5

Lemma 11. *If \mathcal{A} has a non-Zeno run, then in $rGZG^a(\mathcal{A})$ there is an unblocked path visiting a clear node infinitely often.*

Proof. Let ρ be a non-Zeno run of \mathcal{A} :

$$(q_0, \nu_0) \xrightarrow{\delta_0, t_0} (q_1, \nu_1) \xrightarrow{\delta_1, t_1} \dots$$

Since \mathbf{a} is complete, ρ is an instantiation of a path π in $ZG^a(\mathcal{A})$:

$$(q_0, Z_0) \xrightarrow{t_0}_{\mathbf{a}} (q_1, Z_1) \xrightarrow{t_1}_{\mathbf{a}} \dots$$

Let σ be the following sequence of transitions:

$$(q_0, Z_0, Y_0) \xrightarrow{\tau}_{\mathbf{a}} (q_0, Z_0, Y'_0) \xrightarrow{t_0}_{\mathbf{a}} (q_1, Z_1, Y_1) \xrightarrow{\tau}_{\mathbf{a}} (q_1, Z_1, Y'_1) \xrightarrow{t_1}_{\mathbf{a}} \dots$$

where $Y_0 = \text{Rl}(\mathcal{A})$, Y_i is determined by the transition relation in $rGZG^a(\mathcal{A})$, and $Y'_i = Y_i$ unless $\delta_i > 0$ when we put $Y'_i = \emptyset$. We need to see that σ is indeed a path in $rGZG^a(\mathcal{A})$. For this we need to see that every transition $(q_i, Z_i, Y'_i) \xrightarrow{t_i}_{\mathbf{a}} (q_{i+1}, Z_{i+1}, Y_{i+1})$ is realizable from a valuation $\nu \in Z_i$ such that both $\nu \models (\text{Rl}(\mathcal{A}) - Y'_i) > 0$ and $\nu \models g_i$ where g_i is the guard of t_i . We prove this by an induction on the run. As by the definition of ρ , $\nu_i + \delta_i \models g_i$ for all $i \geq 0$, we only need to prove that $\nu_i + \delta_i \models (\text{Rl}(\mathcal{A}) - Y'_i) > 0$. This is clearly true for valuation $\nu_0 + \delta_0 \in Z_0$.

Assume that $\nu_i + \delta_i \models (\text{Rl}(\mathcal{A}) - Y'_i) > 0$. We now prove that $\nu_{i+1} + \delta_{i+1} \models (\text{Rl}(\mathcal{A}) - Y'_{i+1}) > 0$. Firstly, observe that $Y_{i+1} = (Y'_i \cup R_i) \cap \mathcal{C}_0(Z_{i+1}) \cap \text{Rl}(\mathcal{A})$. Therefore a clock $x \in \text{Rl}(\mathcal{A}) - Y_{i+1}$ either belongs to $\text{Rl}(\mathcal{A}) - Y'_i$ in which case it is greater than 0 by induction hypothesis, or otherwise we have $x \in Y'_i$ but $x \notin \mathcal{C}_0(Z_{i+1})$. By the definition of $\mathcal{C}_0(Z_{i+1})$, all valuations $\nu \in Z_{i+1}$ satisfy $\nu(x) > 0$ and so in particular, $\nu_{i+1}(x) > 0$. This leads to $\nu_{i+1} \models (\text{Rl}(\mathcal{A}) - Y_{i+1}) > 0$ which easily extends to $\nu_{i+1} + \delta_{i+1} \models (\text{Rl}(\mathcal{A}) - Y'_{i+1}) > 0$.

Since ρ is non-Zeno there are infinitely many i with $Y'_i = \emptyset$. It is also straightforward to check that σ' is unblocked. \square

Lemma 12. *Suppose $rGZG^a(\mathcal{A})$ has an unblocked path visiting infinitely often a clear node then \mathcal{A} has a non-Zeno run.*

Proof. The proof follows the same lines as the proof of Lemma 6 in [9] with the additional information that for all clocks x that do not belong to $\text{Rl}(\mathcal{A})$, we have $g \wedge (x > 0)$ consistent for all guards g . We recall the proof, with this slight change incorporated.

Let $\pi : (q_0, Z_0, Y_0) \xrightarrow{t_0} \dots$ be the unblocked path of $rGZG^a(\mathcal{A})$ that visit a clear node infinitely often. Since \mathbf{a} is sound, take an instantiation $\rho : (q_0, \nu_0) \xrightarrow{\delta_0, t_0} \dots$ of \mathcal{A} . If ρ is non-Zeno, we are done.

Suppose ρ is Zeno, there exists an index m such that all clocks $\nu_n(x) < 1/2$ for all $x \in X^r$ and for all $n \geq m$. Take indices $i, j \geq m$ such that $Y_i = Y_j = \emptyset$ and

all clocks in X^r are reset between i and j . We look at the sequence $(q_i, \nu_i) \xrightarrow{\delta_i, t_i} \dots (q_j, \nu_j)$ and claim that every sequence of the form

$$(q_i, \nu'_i) \xrightarrow{\delta_i, t_i} (q_{i+1}, \nu'_{i+1}) \xrightarrow{\delta_{i+1}, t_{i+1}} \dots (q_j, \nu'_j)$$

is a part of a run of \mathcal{A} provided there is $\zeta \in \mathbb{R}_{\geq 0}$ such that the following three conditions hold for all $k = i, \dots, j$:

1. $\nu'_k(x) = \nu_k(x) + \zeta + 1/2$ for all $x \notin X^r$,
2. $\nu'_k(x) = \nu_k(x) + 1/2$ if $x \in X^r$ and x has not been reset between i and k .
3. $\nu'_k(x) = \nu_k(x)$ otherwise, i.e., when $x \in X^r$ and x has been reset between i and k .

It is easy to see that the run obtained by replacing every such $i - j$ interval of ρ by the above sequence gives a non-Zeno run, since a $1/2$ time unit has been elapsed infinitely often.

We now show that the above is indeed a valid run of \mathcal{A} . For this we need to first show that $\nu'_k + \delta_k$ satisfies the guard in t_k . Let g be the guard.

For $x \notin X^r$, from the assumption that ρ is unblocked, we know that g could only be of the form $x > c$ or $x \geq c$. So $\nu'_k(x)$ clearly satisfies g . If $x \in X^r$ and is reset between i and k , $\nu'_k(x) = \nu_k(x)$ and so we are done. Consider the case when $x \in X^r$ and is not reset between i and k . Observe that $x \notin Y_k$. This is because $Y_i = \emptyset$, and then only variables that are reset are added to Y . Since x is not reset between i and k , it cannot be in Y_k . By definition of transitions in $rGG^a(\mathcal{A})$, if $x \in \text{RI}(\mathcal{A})$ this means that $g \wedge (x > 0)$ is consistent. But for $x \notin \text{RI}(\mathcal{A})$ by definition, $g \wedge (x > 0)$ is consistent. We have that $0 \leq (\nu_k + \delta_k)(x) < 1/2$ and $1/2 \leq (\nu'_k + \delta_k)(x) < 1$. So $\nu'_k + \delta_k$ satisfies all the constraints in g concerning x as $\nu_k + \delta_k$ does.

It can also be seen that the valuation obtained from ν'_k by resetting the clocks in transition t_k is the valuation ν'_{k+1} . \square

C Proof of Lemma 5

Proof. For the left-to-right direction, suppose that ϕ is satisfiable. Then there exists a variable assignment $\chi : P \mapsto \{\text{true}, \text{false}\}$ that evaluates ϕ to true. We now build the Zeno run of \mathcal{A}_ϕ^Z using χ .

Pick an infinite run ρ of \mathcal{A}_ϕ^Z . Clearly, it should have the following sequence of states repeated infinitely often:

$$q_0 \rightarrow \dots q_k \rightarrow r_0 \rightarrow r_1 \rightarrow \dots r_n \tag{1}$$

We choose the transitions for ρ that allow time elapse only by a finite amount. If $\chi(p_i) = \text{true}$, then we put $q_{i-1} \xrightarrow{\{x_i\}} q_i$ wherever $q_{i-1} \rightarrow q_i$ occurs in ρ . Otherwise $\chi(p_i) = \text{false}$ and we put $q_{i-1} \xrightarrow{\{\overline{x_i}\}} q_i$. We now need to choose the transitions $r_{m-1} \rightarrow r_m$ for $m = 1, \dots, n$. Since χ is a satisfying assignment,

every clause C_m has a literal λ that evaluates to true with χ . We choose the corresponding transition $r_{m-1} \xrightarrow{cl(\bar{\lambda}) \geq 1} r_m$. Observe that if λ evaluates to true, it implies that $cl(\lambda)$ was reset in one of the $q_i \rightarrow q_{i+1}$ transitions but not $cl(\bar{\lambda})$.

Therefore, the above construction yields a sequence of transitions with the property that all clocks that are reset are never checked for greater than 1. This sequence can be taken by elapsing 1 time unit in the very first state, and then subsequently elapsing no time at all, thus giving a Zeno run in \mathcal{A}_ϕ^Z .

We now prove the right-to-left direction. Let ρ be an infinite Zeno run of \mathcal{A}_ϕ^Z . An infinite run should repeat the sequence of states given in (1). Since ρ is Zeno, it has a suffix ρ^s such that for every clock x that is reset in ρ^s , $x \geq 1$ never occurs in the transitions of ρ^s . This is because if every suffix of ρ contains a clock that is both reset and checked for greater than 1, this would mean that there is a time elapse of one time unit occurring infinitely often, contradicting the hypothesis that ρ is Zeno.

Consider a segment $S = q_0 \rightarrow \dots q_n \rightarrow r_0 \rightarrow r_1 \rightarrow \dots r_k$ in ρ^s . We construct a satisfying assignment $\chi : P \mapsto \{true, false\}$ for ϕ from S .

- if S contains $q_{i-1} \xrightarrow{\{x_i\}} q_i$ then set $\chi(p_i) = true$
- otherwise, it implies that S contains $q_{i-1} \xrightarrow{\{\bar{x}_i\}} q_i$ in which case we set $\chi(p_i) = false$.

This shows that for a literal λ , if $cl(\lambda)$ is reset in S , then $\chi(\lambda) = true$. From the property of ρ^s that no clock that is reset is checked in a guard, for every transition $r_{m-1} \xrightarrow{\bar{\lambda} \geq 1} r_m$ in S , it is clock $cl(\lambda)$ that is reset and hence $\chi(\lambda) = true$. By construction of \mathcal{A}_ϕ^Z , λ is a literal in C_m . Therefore, we get a literal that is true in every clause evaluating ϕ to true. \square

D Proof of Theorem 8

Lemma 13. *If \mathcal{A} has a Zeno run, then there exists an infinite slow path in $SZG^a(\mathcal{A})$.*

Proof. Let ρ be a Zeno run of \mathcal{A} :

$$(q_0, \nu_0) \xrightarrow{\delta_0, \tau_0} (q_1, \nu_1) \xrightarrow{\delta_1, t_1} \dots$$

Let π be its concretization in $ZG^a(\mathcal{A})$:

$$(q_0, Z_0) \xrightarrow{t_0}_a (q_1, Z_1) \xrightarrow{t_1}_a \dots$$

We construct an infinite slow path in $SZG^a(\mathcal{A})$ from the path π . Let X^l be the set of clocks that are lifted infinitely often in π and let X^r be the set of clocks that are reset infinitely often in π . Let π^i denote the suffix of π starting from the position i .

Clearly, there exists an index m such that all the clocks that are lifted in π^m belong to X^l and the ones that are reset in π^m belong to X^r . Since ρ is Zeno, we have $X^l \cap X^r = \emptyset$. This shows that all the clocks that are reset in π^m are never lifted in its transitions. Therefore, there exists an index $k \geq m$ such that for all $j \geq k$, Z_j is consistent with $x < 1$ for all clocks $x \in X^r$ and we get the following path of $SZG^a(\mathcal{A})$:

$$(q_0, Z_0, \text{free}) \xRightarrow{t_0}_a \dots (q_j, Z_j, \text{free}) \xRightarrow{\tau}_a (q_j, Z_j, \text{slow}) \xRightarrow{t_j}_a (q_{j+1}, Z_{j+1}, \text{slow}) \xRightarrow{t_{j+1}}_a \dots$$

□

Lemma 14. *If $SZG^a(\mathcal{A})$ has an infinite slow path, then \mathcal{A} has a Zeno run.*

Proof. Let π be the slow path of $SZG^a(\mathcal{A})$:

$$(q_0, Z_0, \text{free}) \xRightarrow{t_1}_a \dots (q_j, Z_j, \text{free}) \xRightarrow{\tau}_a (q_j, Z_j, \text{slow}) \xRightarrow{t_j}_a (q_{j+1}, Z_{j+1}, \text{slow}) \xRightarrow{t_{j+1}}_a \dots$$

Take the corresponding path in $ZG^a(\mathcal{A})$ and an instance $\rho = (q_0, \nu_0) \xrightarrow{\delta_0, t_0} (q_1, \nu_1) \dots$ which is a run of \mathcal{A} , as we have assumed that a is a sound abstraction.

Let X^r be the set of clocks that are reset infinitely often and let X^l be the set of clocks that are lifted infinitely often in ρ . By the semantics of the slow mode and from our hypothesis of a being lift-safe, after the index j , all clocks that are lifted once can never be reset again. Therefore, there exists an index $k \geq j$ such that the following hold:

- all clocks that are reset in ρ^k belong to X^r and all clocks that are lifted in a transition of ρ^k belong to X^l ,
- for all $x \in X^l$ and for all $i \geq k$, $\nu_i(x) \geq c$ where c is the maximum constant appearing in a lifting transition of ρ^k .

We now modify the time delays of ρ^k to construct a run that elapses a bounded amount of time. Pick the sequence of indices i_1, i_2, \dots in ρ^k such that $\delta_{i_m} > 0$, for all $m \in \mathbb{N}$. Define the new delays δ'_i for all $i \geq k$ as follows:

$$\delta'_i = \begin{cases} \min(\delta_i, \frac{1}{2^j}) & \text{if } i = i_j \text{ for some } j \\ 0 & \text{otherwise} \end{cases}$$

Consider the run ρ' obtained by elapsing δ'_i time units after the index k :

$$(q_0, \nu_0) \xrightarrow{\delta_0, t_0} \dots \xrightarrow{\delta_{k-1}, t_{k-1}} (q_k, \nu_k) \xrightarrow{\delta'_k, t_k} (q_{k+1}, \nu'_{k+1}) \xrightarrow{\delta'_{k+1}, t_{k+1}} \dots$$

Clearly, ρ' is Zeno. It remains to prove that ρ' is a run of \mathcal{A} . Denote ν_k by ν'_k . We need to show that for all $i \geq k$, $\nu'_i + \delta'_i$ satisfies the guard in the transition t_i . Call this guard g_i . Clearly, since $\nu'_i + \delta'_i \leq \nu_i + \delta_i$ by definition, if g_i is of form $x < c$ or $x \leq c$ then it is satisfied by the new valuation. Let us now consider the

case when g_i is of the form $x \geq c$ or $x > c$. If $c \geq 1$, then we know that $x \in X^l$ from the assumption on k . But since $\nu_k(x) \geq c$ and x is not reset anywhere in ρ^k , $\nu'_i(x) \geq c$ for all i and hence the new valuation satisfies g_i . We are left with the case when g_i is of the form $x > 0$. However this follows since by definition of the new δ'_i , $\nu'_i + \delta'_i = 0$ iff $\nu_i + \delta_i = 0$. \square